Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение   
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**РАЗРАБОТКА ПРИЛОЖЕНИЯ NOTEAPP**

Пояснительная записка к проекту, разработанному по дисциплине   
«Новые технологии в программировании»

Студент гр. 588-2

\_\_\_\_\_\_\_\_\_\_\_А.Д. Целицкий

« » 20\_\_

Руководитель

доцент кафедры КСУП

\_\_\_\_\_\_\_\_\_\_\_ А.Е. Горяинов  
 « » 20\_\_

Томск 2022

Содержание

[1 Назначение приложения 3](#_Toc75107228)

[2 Группы пользователей и их функциональные возможности в приложении 4](#_Toc75107229)

[3 Стек технологий разработки и системные требования 5](#_Toc75107230)

[4 Пользовательский интерфейс 6](#_Toc75107231)

[5 Диаграмма пакетов приложения 13](#_Toc75107232)

[6 Диаграмма классов приложения 14](#_Toc75107233)

[7 Описание тестирования приложения 15](#_Toc75107234)

[8 Описание сборки установщика 18](#_Toc75107235)

[9 Описание модели ветвления 22](#_Toc75107236)

# Назначение приложения

Пользовательское приложение NoteApp предназначено для ведения персональных записей и заметок. Приложение должно:

1. Обеспечивать стабильную работу приложения при порядке 200 заметок.
2. Обеспечивать категоризацию заметок, навигацию по созданным заметкам.
3. Предоставить инструменты для просмотра и редактирования заметок.
4. Сохранять и восстанавливать заметки между сессиями приложения.
5. Выполнять промежуточные сохранения заметок на машине пользователя на случай аварийного завершения программы, отключения компьютера и т.д. для защиты от потери данных.

Приложение-референс: десктоп-версия программы Evernote.

# Группы пользователей и их функциональные возможности в приложении

В данном приложении роль пользователя единственная. Пользователь приложения имеет следующие функциональные возможности:

1. создание заметки;
2. редактирование заметки;
3. удаление заметки;
4. просмотр ранее созданных заметок;
5. просмотр информации о программе.

Также пользователь может выбрать одну из следующих категорий заметки: «Все», «Работа», «Дом», «Здоровье и Спорт», «Люди», «Документы», «Финансы», «Разное» – для просмотра заметок, относящихся только к данной категории.

# Стек технологий разработки и системные требования

Для реализации проекта был выбран язык программирования С# на платформе .NET 4.7.2, набор библиотек Windows Forms для создания десктоп-приложений под операционной системой Windows, а также среда разработки Microsoft Visual Studio Professional 2019 версии 16.10.2 с применением Microsoft .NET Framework 4.8. Все системные требования обусловлены требованиями к платформе .NET.

Для обеспечения сериализации и десериализации данных приложения была выбрана библиотека Newtonsoft JSON.NET версии 12.0.3.

Для проведения юнит-тестирования проекта бизнес-логики приложения в среде разработки Visual Studio 2019 была использована библиотека NUnit с пакетами NUnit версии 3.13.2 (непосредственно фреймворк для автоматизации тестирования) и NUnit3TestAdapter версии 4.0.0 (дополнительный пакет, позволяющий запускать юнит-тесты в среде Visual Studio). Выбор данной библиотеки обоснован достаточной гибкостью и удобством написания юнит-тестов при помощи данной среды.

Для создания сценария сборки установочного пакета и компиляции установщика приложения была выбрана программа сборки установочных пакетов InnoSetup версии 6.2.0.

# Пользовательский интерфейс

После запуска приложения перед пользователем появляется главное окно (рисунок 1). Двухколоночная верстка главного окна содержит список всех созданных заметок в левой панели и отображает текущую выбранную заметку в правой панели. В списке заметок показаны названия заметок, в один момент времени может быть выбрана только одна заметка (далее – текущая заметка).

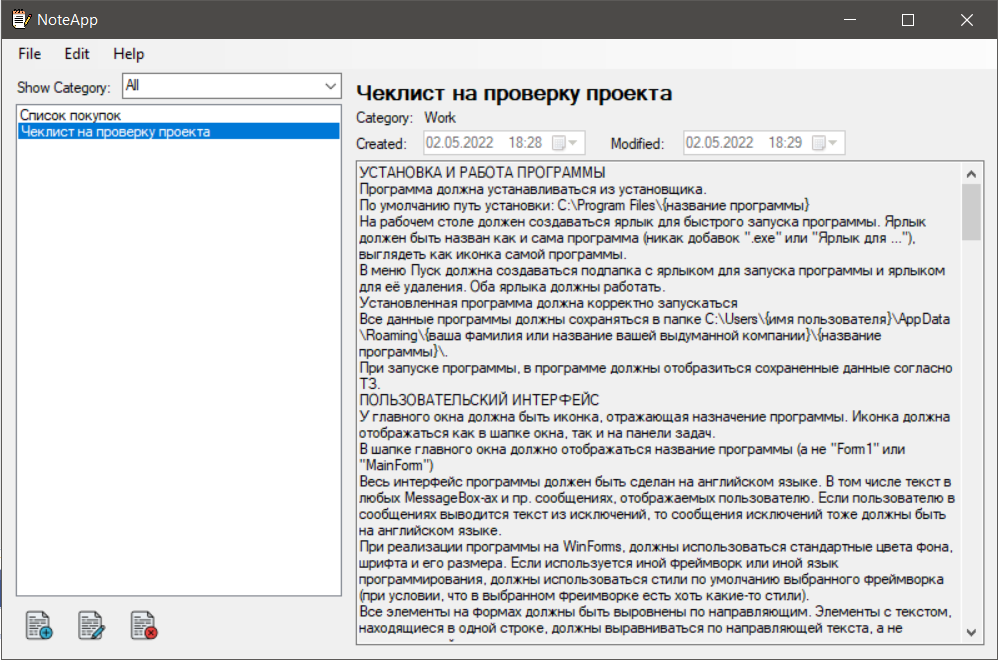


Рисунок 1 – Главное окно программы

Список заметок в левой панели отсортирован по дате изменения: последние измененные записи находятся вверху списка, а записи с более ранними датами изменения – внизу.

На панели со списком заметок вверху расположен выпадающий список категорий заметок. При выборе пользователем определенной категории заметки в списке заметок отображаются только заметки выбранной категории. При выборе пункта «All» отображаются все заметки приложения.

Меню главного окна содержит следующие пункты:

* *File*
* *Exit* (Выйти из приложения – Alt+F4)
* *Edit*
* *Add Note* (Создать новую заметку)
* *Edit Note* (Редактировать текущую заметку)
* *Remove Note* (Удалить текущую заметку – Del)
* *Help*
* *About* (Вызвать окно «О программе» – F1)

На панели со списком заметок внизу располагаются три кнопки: Add Note («Создать новую заметку»), Edit Note («Редактировать текущую заметку»), Delete Note («Удалить текущую заметку»). При наведении курсора мыши на каждую из данных кнопок появляется всплывающая подсказка, которая кратко описывает назначение интересующей пользователя кнопки.

При выборе заметки из списка в левой панели главного окна выбранная заметка отображается в его правой панели. Главное окно не позволяет редактировать содержимое заметки – возможен только просмотр.

При нажатии на кнопку Add Note и Edit Note появляется окно создания/редактирования заметки в диалоговом режиме (рисунок 2). Для новой заметки окно изначально не заполнено, установлены только название заметки по умолчанию, категория по умолчанию, дата создания и дата редактирования. При создании заметки в категории «All» категория по умолчанию для новой заметки – «Documents»; если же заметка создается при какой-либо другой выбранной категории, то эта категория изначально присваивается новой заметке. Для редактирования уже существующей заметки все поля заполняются данными текущей заметки.

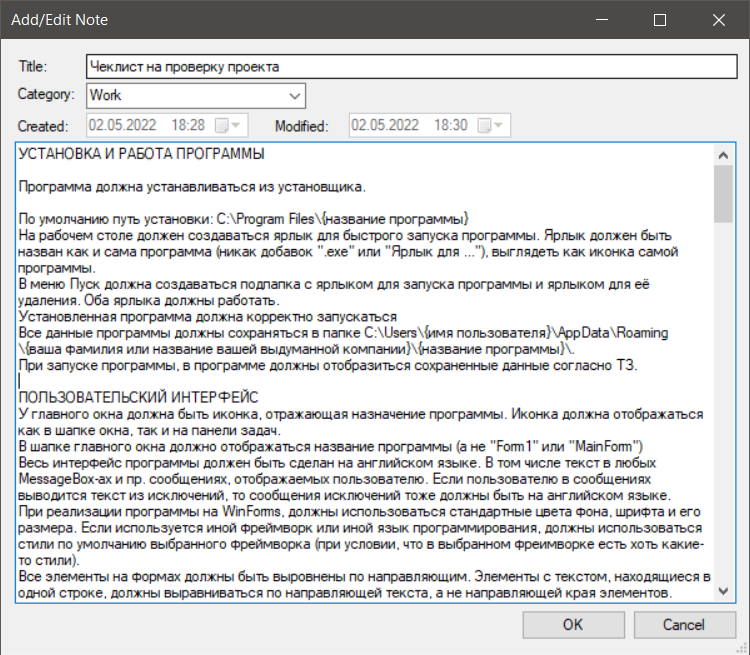


Рисунок 2 – Окно создания и редактирования заметки

При нажатии на кнопку OK окно создания заметки закрывается, в список заметок главного окна добавляется новая заметка. При редактировании текущей заметки нажатие на кнопку OK обновляет название заметки в списке заметок, а также обновляет отображаемую заметку в правой панели приложения. При нажатии кнопки Cancel создание/редактирование заметки отменяется (новая заметка не добавляется, исходная заметка остается без изменений).

В случае ввода пользователем некорректных данных (нарушения допустимой длины названия заметки) данная ситуация обрабатывается следующим образом: элемент управления с названием заметки (поле ввода) подсвечивается красным фоном, при наведении курсора мыши на поле появляется всплывающая подсказка с текстом ошибки («Title length should not exceed 50 characters»), как показано на рисунке 3.

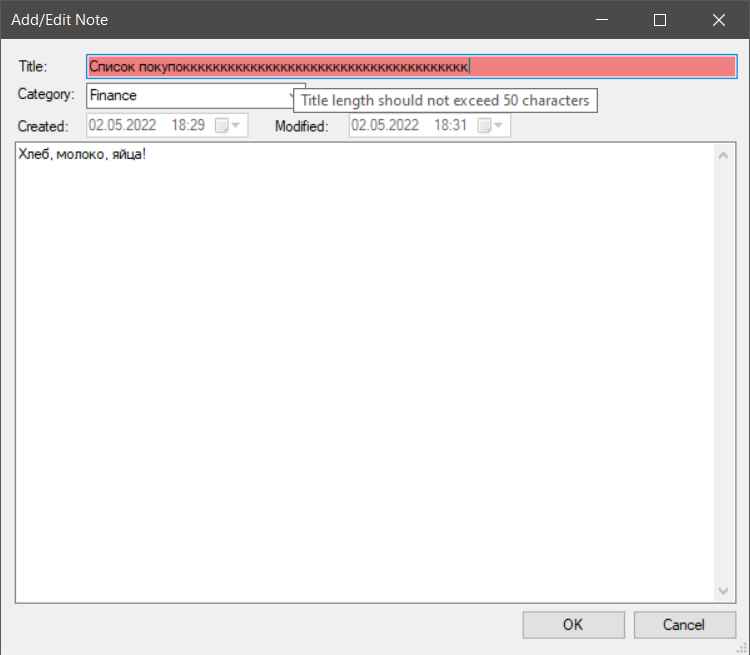


Рисунок 3 – Демонстрация обработки ввода некорректного названия заметки

В случае если пользователь проигнорирует предупреждение о превышении длины названия заметки и попытается, нажав на кнопку «ОК», сохранить изменения в заметке, содержащей некорректные данные, появится окно с перечислением всех данных, которые ему необходимо исправить для возможности сохранения заметки (рисунок 4).

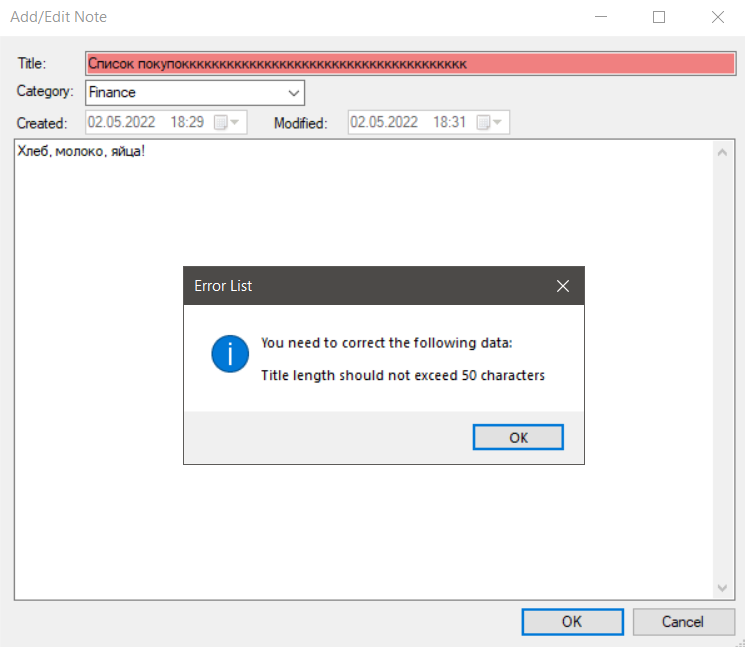


Рисунок 4 – Демонстрация запрета сохранения изменений в заметке, содержащей ошибки пользовательского ввода

В случае исправления пользователем недопустимого названия элемент управления с названием заметки становится корректным, а всплывающая подсказка с текстом ошибки исчезает.

При нажатии на кнопку Remove Note главного окна текущая запись удаляется. Перед удалением появляется окно с запросом на разрешение удаления: («Do you really want to remove this note: <Название текущей заметки>?»), как показано на рисунке 5.

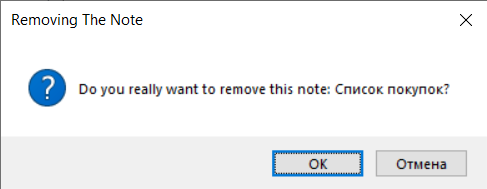


Рисунок 5 – Окно с запросом на разрешение удаления текущей заметки

При нажатии на кнопку OK происходит удаление текущей заметки, при нажатии на кнопку Cancel удаление отменяется. Также удаление заметки можно произвести по нажатию на клавишу Delete.

Информацию о приложении пользователь может просмотреть, вызвав информационное окно «About» из меню главного окна или нажатием на клавишу F1 (рисунок 6).

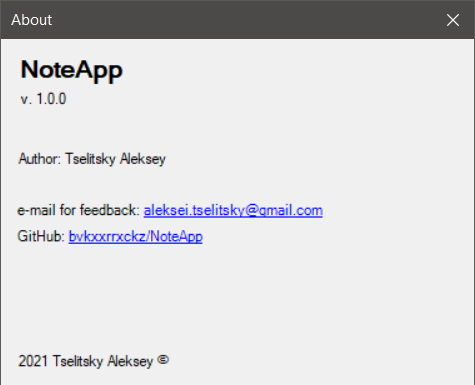


Рисунок 6 – Окно «About»

Таким образом, в приложении реализовано три окна:

- главное окно;

- окно создания/редактирования заметки;

- окно «About».

Верстка главного окна и окна создания/редактирования заметки адаптивны. Окно «About» имеет фиксированный размер.

Загрузка заметок осуществляется при запуске программы до вывода главного окна пользователю, сохранение заметок в файл выполняется в случаях: а) создания новой заметки; б) удаления заметки; в) закрытия приложения.

Также в приложении запоминается текущая заметка. При закрытии приложения индекс текущей заметки сохраняется в файл, а при запуске – загружается из файла, что позволяет при повторном открытии приложения отобразить пользователю последнюю просмотренную им заметку.

# Диаграмма пакетов приложения

Диаграмма пакетов отображает архитектуру приложения, разделенную на отдельные пакеты – библиотеки. Внутри библиотек описываются доступные извне классы, между пакетами рисуются направленные линии, обозначающие связи между библиотеками.

Диаграмма пакетов приложения представлена на рисунке 7.

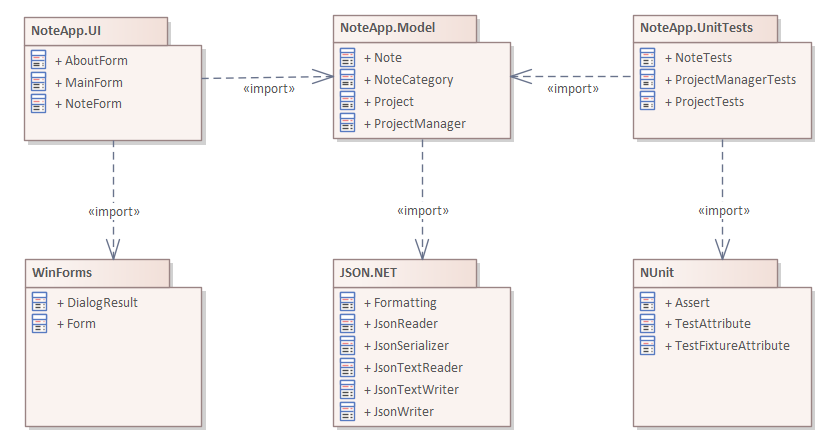


Рисунок 7 – Диаграмма пакетов приложения

# Диаграмма классов приложения

Проект бизнес-логики приложения содержит следующие типы данных:

1. класс Note – хранит информацию о названии, категории, тексте, времени создания и времени последнего изменения заметки;
2. перечисление NoteCategory – хранит категории заметки;
3. класс Project – содержит список всех заметок, созданных в приложении;
4. класс ProjectManager – реализует метод для сохранения объекта Project в файл и метод загрузки объекта Project из файла.

Диаграмма классов приложения приведена на рисунке 8.

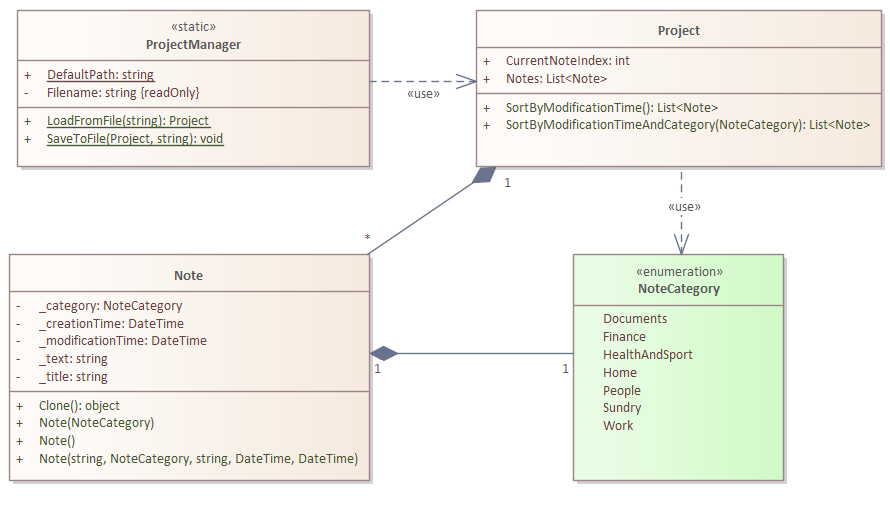


Рисунок 8 – Диаграмма классов приложения

# Описание тестирования приложения

Тестирование приложения было проведено как вручную, так и автоматически.

Основным типом тестирования было юнит-тестирование. С помощью юнит-тестов в разработанном приложении протестирован проект бизнес-логики. Для этого разработаны тесты, полностью покрывающие исходный код данного проекта. Информация о результатах тестирования приведена на рисунке 9.

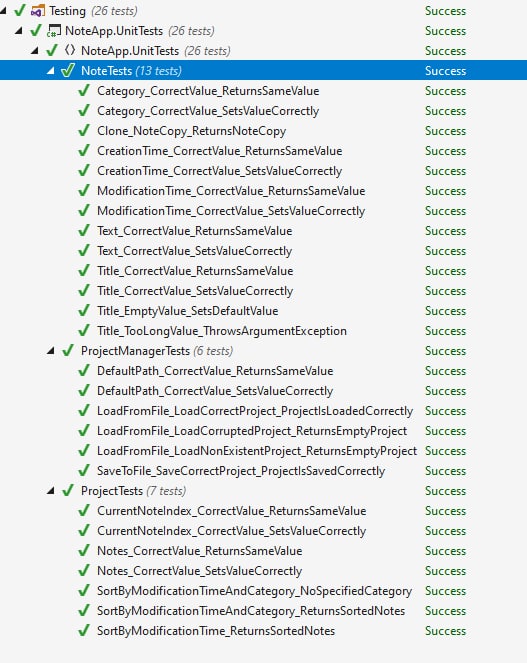


Рисунок 9 – Результаты тестирования проекта бизнес-логики

Тесты написаны с использованием библиотеки NUnit, предназначенной для написания юнит-тестов для платформы .NET. Написанные тесты проверяют как позитивные, так и негативные тестовые случаи.

Тестирование классов пользовательского интерфейса проводилось вручную.

В качестве общего системного теста, проверяющего работоспособность всего приложения, было проведено приемочное тестирование. Для приемочного тестирования необходимо выполнить следующую последовательность действий:

1. Установить приложение на компьютер с помощью собранного установочного пакета.
2. Запустить приложение. Окно программы должно быть пустым – в приложении не должно быть заметок.
3. Создать в приложении три заметки разных категорий.
4. Переключиться между заметками, чтобы убедиться, что смена текущей заметки происходит корректно.
5. Переключить отображаемую категорию заметок – в списке заметок должны остаться только заметки целевой категории. Снова отобразить все категории заметок – список заметок должен восстановиться.
6. Выбрать вторую заметку и нажать кнопку редактирования. Должно открыться окно редактирования заметки.
7. Ввести название заметки более 50 символов. Элемент управления с названием заметки должен указать на некорректное значение.
8. Ввести название заметки менее 50 символов. Элемент управления с названием заметки должен стать корректным.
9. Поменять текст заметки. Нажать «OK». Отредактированная заметка должна подняться в списке заметок на первую позицию, время изменения заметки должно поменяться, отображаемый текст заметки также должен измениться.
10. Выбрать вторую заметку и нажать кнопку редактирования. Должно открыться окно редактирования. Изменить название заметки, ее текст и категорию. Нажать «Cancel». Исходная заметка должна остаться без изменений.
11. Удалить третью заметку.
12. Закрыть приложение. Должно произойти сохранение заметок в целевой файл.
13. Запустить приложение. В программе должны восстановиться заметки, созданные в предыдущую сессию.
14. Дать руководителю провести исследовательское тестирование программы.

# Описание сборки установщика

Для сборки установочного пакета приложения и компиляции установщика используется программа InnoSetup. С ее помощью создан сценарий для сборки установочного пакета. Сценарии сохраняются в формате \*.iss (inno setup scripts) и имеют специальный синтаксис.

Сценарий сборки содержит всю необходимую для установки приложения информацию, пути установки, название установочного пакета, а также может содержаться код для создания дополнительных, нестандартных окон в установщике. В сценарии необходимо указать файлы, которые требуются для работы приложения, такие как файлы с расширениями \*.exe и \*.dll. Остальные файлы (\*.pdb, \*.config, \*.manifest, \*.xml и др.) для работы приложения не нужны и в установочный пакет не входят.

Далее представлен сценарий сборки установочного пакета разработанного приложения.

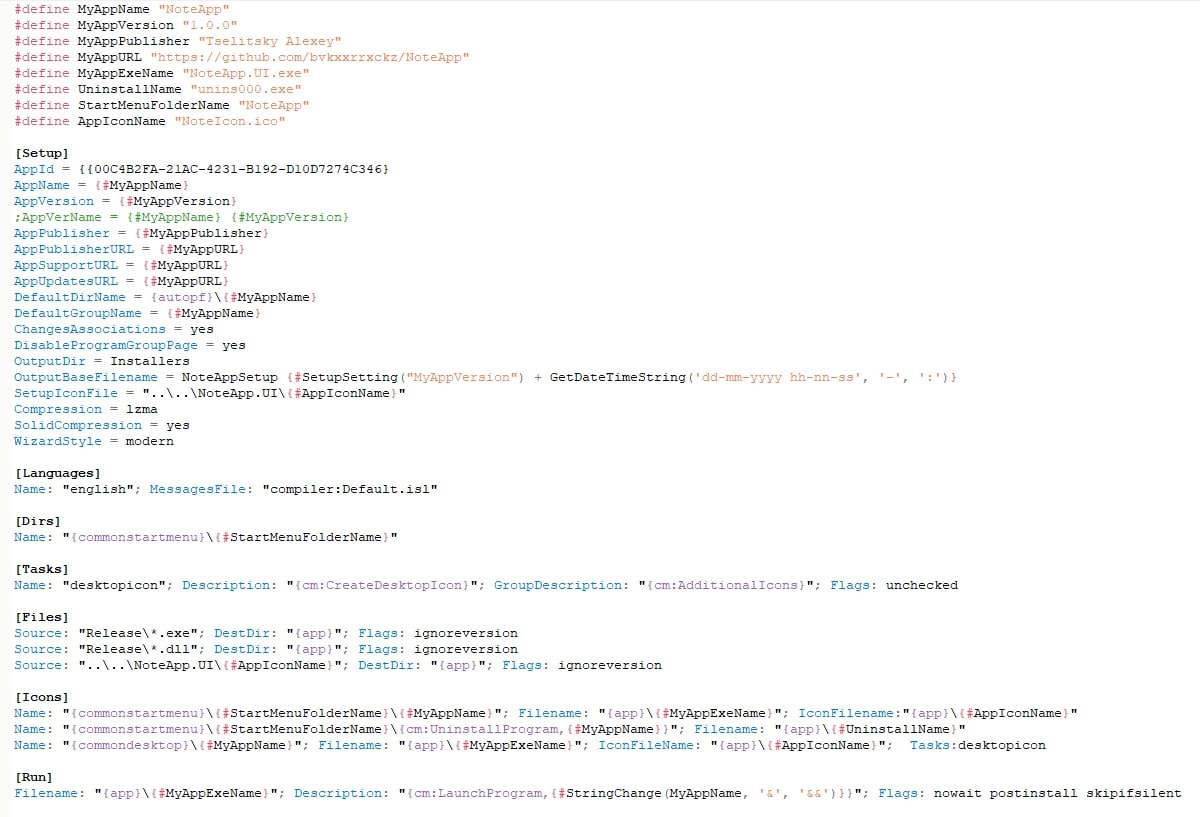


Рисунок 10 – Сценарий сборки установочного пакета

Запуск сценария сборки происходит в среде Visual Studio. Сборка установочного файла запускается автоматически во время компиляции приложения. Файл сценария помещен в папку решения. Для автоматизации сборки установщика сделаны дополнительные настройки в решении: в настройки проекта пользовательского интерфейса NoteApp.UI добавлены следующие команды, выполняющиеся после каждой успешной компиляции:

rmdir /s /q "$(SolutionDir)Installer\InstallScripts\Release"

md "$(SolutionDir)Installer\InstallScripts\Release"

md "$(SolutionDir)Installer\InstallScripts\Installers"

xcopy "$(ProjectDir)$(OutDir)\*.exe"

"$(SolutionDir)Installer\InstallScripts\Release"

xcopy "$(ProjectDir)$(OutDir)\*.dll"

"$(SolutionDir)Installer\InstallScripts\Release"

где:

rmdir – удаление папки;

md – создание новой папки;

xcopy – копирование из одной папки в другую.

Чтобы после компиляции проекта автоматически запускалась компиляция установочного пакета согласно сценарию Installer.iss, в настройки проекта по сборке установщика Installer добавлена следующая команда, которая выполняется после каждой успешной компиляции:

"$(SolutionDir)packages\Tools.InnoSetup.6.2.0\tools\ISCC.exe" "$(SolutionDir)Installer\InstallScripts\Installer.iss"

В результате выполнения сценария сборки установочного пакета был получен установочный файл NoteAppSetup.exe, при запуске которого происходит установка приложения NoteApp на машину пользователя.

# Описание модели ветвления

Во время разработки приложения для обеспечения версионного контроля использовался сервис GitHub.

GitHub.com – это один из крупнейших веб-сервисов для хостинга   
IT-проектов и их совместной разработки. GitHub основан на системе контроля версий Git. Данный веб-сервис позволяет хранить промежуточные версии исходного кода. Таким образом, в случае написания непоправимых ошибок в исходном коде или его потери сервис позволит восстановить проект. Также GitHub является обязательным инструментом для организации командной работы, используемым во многих IT-компаниях. Предоставляет бесплатную лицензию для малых проектов.

Ссылка на репозиторий: https://github.com/AkhromenkoDarya/NoteApp.git.

Разработка приложения проводилась в двух ветках репозитория:

1. main – главная ветвь, предназначенная для хранения только тех коммитов, код которых проверен, протестирован и готов для сборки установщика;
2. develop – специальная ветка для непосредственной разработки программы, которая создается из ветки main сразу же после создания репозитория и существует все время разработки проекта.

Фиксации изменений в репозитории производились из интегрированной среды разработки Visual Studio.